MSX LEERBOEK

OPDRACHTEN BIJ

DEEL 2

WESSEL AKKERMANS/PIET DEN HEIJER

Vragen en opdrachten

1 Standaard MSX-functies en eigen functies

1-1:
Met welke functie kan de absolute waarde van een getal worden bepaald?

1-2:
Schrijf een programma, waarmee de absolute waarde van het resultaat van de berekening X-Y wordt bepaald. De waarden voor X en Y dienen met INPUT-statements te worden ingevoerd.

1-3:
Wat is het verschil tussen de functie FIX(X) en de functie INT(X)?

1-4:
Schrijf een programma, waarmee van het getal X het gehele deel aan numerieke variabele G wordt toegekend en het gedeelte achter de komma (decimale punt) aan numerieke variabele B. Het getal X (positief of negatief) dient te worden ingevoerd met een INPUT-statement. Verder moeten de variabelen G en B worden afgedrukt op het scherm.

1-5:
Wat is de ASCII-code (decimale waarde) voor het besturingsteken "Carriage Return". Hoe kan in een programma "Carriage Return" worden geactiveerd?

1-6:
Schrijf een programma, waarmee de volgende berekening kan worden uitgevoerd:

$$y = \sqrt{(a+b)/c}$$

De variabelen a, b en c dienen met INPUT-statements te worden ingevoerd. Het resultaat y moet op het scherm worden afgedrukt. Het programma moet, na het uitvoeren van de berekening en het afdrukken van het resultaat, door het indrukken van de RETURN-toets weer kunnen terugspringen naar

het begin van het programma, zodat er weer nieuwe waarden kunnen worden ingevoerd voor de variabelen a, b en c.

1-7:
Met welke functie kunnen willekeurige getallen worden gegenereerd?

1-8:
Schrijf een programma, waarmee 12 willekeurige gehele getallen, die liggen tussen 0 en 21, worden gegenereerd. (0 en 21 behoren ook tot de reeks.)

1-9:
Pas het programma uit 1-8 zodanig aan, dat er 12 getallen tussen 1 en 21 worden gegenereerd.

1-10:
Pas het programma uit 1-9 zodanig aan, dat de waarde van het grootste getal, dat mag worden gegenereerd, door middel van een INPUT-statement wordt ingevoerd. Laat het programma verder, door middel van het indrukken van de RETURN-toets, terugspringen naar het begin van het programma.

1-11:
Maak in het programma uit vraag 1-10 ook het aantal te genereren getallen variabel. De eindwaarde dient hiervoor met een INPUT-statement te worden ingevoerd.

1-12:
Met welk statement kan een eigen functie worden gedefinieerd?

1-13:
Stel de programma regel samen voor het definieren van de functie, waarmee de oppervlakte van een driehoek wordt berekend. Noem in de formule de basis B en de hoogte H. Geef de functie de naam OPDR.

1-14:
Schrijf een programma, waarmee de oppervlakte van driehoeken kan worden berekend. Gebruik in het programma de functiedefinitie uit vraag 1-13. De waarden voor de variabelen B en H dienen met INPUT-statements te worden ingevoerd. Verder moet het resultaat op het beeldscherm worden afgedrukt.

1-15:
Schrijf een programma voor het berekenen van de inhoud van cilinders. De formule hiervoor is: V=(3.14*R^2)*H. Verwerk de formule in een eigen functie die de naam INHC krijgt.

1-16:
Hoe worden de variabelen in een DEF-statement ook wel genoemd?

1-17:
Moeten de variabelen in een DEF-statement dezelfde naam dragen als in de uit te voeren functie (FN)?

1-18:
Schrijf een programma, waarmee de volgende berekening wordt gemaakt:

$$R = \sqrt{(X+Y)^2/Z}$$

Noem de variabelen in het DEF-statement X, Y en Z en in de uit te voeren functie respectievelijk A, B en C. De variabelen A, B en C dienen met 1 INPUT-statement te worden ingevoerd. Het resultaat moet worden afgedrukt op het scherm. Geef de functie de naam FUN.

2 Afdrukken op scherm en papier

2-1:
Met welk commando kan men het programma, dat in het geheugen staat, afdrukken op papier?

2-2:
Hoe kan de laatst behandelde programmaregel worden afgedrukt op papier?

2-3:
Verander, in de directe mode, de functie van functietoets F2 zodanig, dat na het indrukken van de toets het op dat moment in het geheugen aanwezige programma wordt afgedrukt op papier.

2-4:
Schrijf een programma, waarmee van een reeks getallen, de getallen zelf (I), de kwadraten van de getallen (I^2) en 1 gedeeld door de getallen (1/I) op papier worden afgedrukt. Het eerste getal (E) en het laatste getal (L) van de reeks dienen te worden ingevoerd met INPUT-statements. Druk na het uitvoeren van het programma ook het programma zelf af op papier.
Opmerking: De waarde van L mag niet groter zijn dan 1000.

2-5:
Wat geeft de systeemvariabele CSRLIN aan en welke waarden kunnen door het systeem aan de variabele worden toegekend?

2-6:
In welke modes kan CSRLIN worden toegepast?

2-7:
Schrijf een programma, waarmee van een reeks opeenvolgende getallen, de getallen zelf en de kwadraten van de getallen worden afgedrukt op het beeldscherm. De beginwaarde van de reeks moet worden ingevoerd met een INPUT-statement.
Wanneer de tabel regel 22 op het scherm heeft bereikt, moet het programma wachten met het verder afdrukken van de tabel, totdat de RETURN-toets wordt ingedrukt. Hierna moet het afdrukken weer worden voortgezet. Alvorens het afdrukken wordt voortgezet, moet eerst het beeldscherm worden schoongemaakt.

2-8:
Wat geeft de systeemvariabele POS(0) aan en welke waarden kunnen door het systeem aan de variabele worden toegekend?

2-9:
Schrijf een programma, waarmee de letters A tot en met Z op het beeldscherm worden afgedrukt. Tussen elke twee letters moeten twee spaties worden geplaatst. Nadat positie 20 op de regel is bereikt, moet worden doorgegaan op de volgende regel.

2-10:
Wat wordt aangegeven met de systeemvariabele LPOS(0)?

2-11:
Pas het programma uit vraag 2-9 zodanig aan, dat de letters A tot en met Z op papier worden afgedrukt.

3 Goniometrische functies

3-1:
Welke standaardfuncties kent MSX-BASIC voor de goniometrie?

3-2:
Hoe kan argument X voor de verschillende functies worden uitgedrukt?

3-3:
Hoe wordt het aantal radialen (X) berekend, als het aantal graden (G) is gegeven?

3-4:
Schrijf een programma, waarmee graden worden omgerekend in radialen. De graden (G) moeten worden ingevoerd met een INPUT-statement. Het resultaat dient te worden afgedrukt op het scherm. Het programma moet, na het indrukken van de RETURN-toets, weer terugspringen naar het begin van het programma.

3-5:
Schrijf een hoofdprogramma, waarmee de cosinus van een hoek wordt berekend. De hoek dient in een subroutine te worden ingevoerd en te worden omgezet in radialen. Het berekenen van de cosinus en het afdrukken van het resultaat, in 5 cijfers nauwkeurig achter de komma, moet in het hoofdprogramma geschieden.

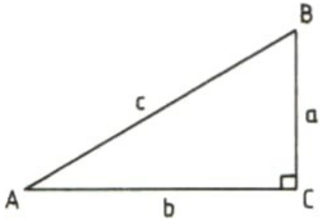
3-6:
Verander de subroutine uit vraag 3-5 zodanig, dat er ook hoeken kunnen worden ingevoerd, die in graden, minuten en seconden zijn uitgedrukt.

3-7:
Schrijf een programma, waarmee een sinus en cosinustabel op het scherm kan worden afgedrukt. De tabel moet worden opgebouwd uit drie kolommen. In de eerste kolom komt het aantal graden (G) te staan, in de tweede kolom de sinus van de hoek G en in de derde kolom de cosinus van de hoek G. De sinus en cosinus moeten in vijf cijfers nauwkeurig achter de komma worden afgedrukt. De beginwaarde B en de eindwaarde E, uitgedrukt in graden, moeten met INPUT-statements worden ingevoerd. De waarde in kolom 1 moet steeds met 1 graad worden opgehoogd. Wanneer regel 20 op het scherm is bereikt,

moet het programma stoppen met afdrukken, en pas weer verder gaan met het afdrukken van de tabel, wanneer de RETURN-toets wordt ingedrukt. Na het indrukken van de RETURN-toets moet eerst het scherm worden schoongemaakt.

3-8:
Wat is de relatie tussen de functie ATN(X) en TAN(X)? Hoe wordt een hoek (G), uitgedrukt in radialen, omgezet naar graden?

3-9:
Schrijf een programma, waarmee met de functie ATN(X) hoek A wordt berekend. De lengte van de zijden a en b dienen te worden ingevoerd met INPUT-statements. De berekende hoek moet worden afgedrukt in graden (twee cijfers nauwkeurig achter de komma).



3-10:
Wat wordt er met de functie EXP(X) berekend?

3-11:
Schrijf een programma, waarmee een tabel wordt afgedrukt met de resultaten van de functie EXP(X). De beginwaarde is X=0.1, de eindwaarde is X=1.9 en de stapwaarde is 0.1. In de tabel moeten zowel de waarde van X als de resultaten van de functie worden afgedrukt.

3-12:
Wat wordt er met de functie LOG(X) berekend? Hoe kan argument X worden uitgedrukt?

3-13:
Hoe kan, door middel van de natuurlijke logaritme, de normale of Briggse logaritme worden berekend?

3-14:
Schrijf een subroutine, waarmee de natuurlijke logaritme van een getal kan worden berekend. Het getal dient te worden ingevoerd door een INPUT-statement in de subroutine. Het resultaat dient eveneens te worden afgedrukt door de subroutine. De subroutine start op regelnummer 200.

3-15:
Schrijf een subroutine, waarmee de Briggse logaritme van een getal kan worden berekend. Het getal dient te worden ingevoerd door een INPUT-statement in de subroutine. Het resultaat dient eveneens te worden afgedrukt door de subroutine. De subroutine start op regelnummer 300.

3-16:
Schrijf een programma, waarmee vanuit een menu kan worden gekozen voor het berekenen van de natuurlijke logaritme of de Briggse logaritme van een getal en voor het beeindigen van het programma. Gebruik hiervoor het statement ON K GOSUB en de subroutines uit de vragen 3-14 en 3-15. Het programma start op regelnummer 100.

4 De grafische macrotaal

4-1:
Welk statement wordt gebruikt in de grafische macrotaal?

4-2:
In welke beeldschermmodes mag het statement uit vraag 4-1 alleen worden gebruikt?

4-3:
Hoe is het scherm voor de grafische modes ingedeeld? (x- en y-coordinaten.)

4-4:
Wat is de kleinste eenheid voor beide modes en hoe nauwkeurig kunnen de eenheden worden gepositioneerd?

4-5:
Welk subcommando wordt gebruikt voor het trekken van een lijn?

4-6:
Welk subcommando kan worden gebruikt, wanneer men niet weet waar de cursor zich bevindt en men de cursor eerst op een bepaalde positie wil plaatsen? Welke twee subcommando's worden gebruikt om de cursor linksonder op het beeldscherm te plaatsen?

4-7:
Schrijf een programma, waarmee in de grafische mode 1 op het scherm een lijn wordt getrokken van linksonder naar rechtsboven. Het is niet bekend waar de cursor staat.

4-8:
Breid het programma uit vraag 4-7 zodanig uit, dat er ook een lijn wordt getrokken van linksboven naar rechtsonder.

4-9:
Hoe kan de cursor, na het trekken van een lijn, weer worden teruggeplaatst naar het beginpunt?

4-10:
Wat moet er in de grafische modes eerst worden gedaan, voordat er op het scherm teksten kunnen worden geschreven? Hoe zie het formaat van het betreffende statement er uit?

4-11:
Breid het programma uit vraag 4-8 zodanig uit, dat in de vier ontstane velden respectievelijk de letters A, B, C en D worden afgedrukt.

4-12:
Schrijf een programma, waarmee met het subcommando Mx,y een rechthoek op het scherm wordt getekend van 90 beeldpuntjes lang en 60 beeldpuntjes hoog. Het startpunt ligt op x=80 en y=60 (hoek linksonder).

4-13:
Hoe weet het programma of er met absolute of relatieve waarden moet worden gewerkt?
Opgave:
Er moet vanuit punt x=80 en y=70 een lijn worden getrokken naar x=120 en y=90. Hoe gaat het subcommando er uitzien voor absolute waarden en hoe voor relatieve waarden?

4-14:
Pas het programma uit vraag 4-12 aan voor relatieve waarden.

4-15:
Met welke subcommando's kunnen lijnen worden getrokken in acht verschillende richtingen?

4-16:
Schrijf een programma, waarmee vanuit het middelpunt van het beeldscherm (x=128 en y=91) lijnen worden getrokken in acht verschillende richtingen. Het argument n=60. Na het trekken van elke lijn moet de cursor weer terug worden geplaatst naar het middelpunt.

4-17:
Schrijf een programma, waarmee een zeskant wordt getekend. Gebruik hiervoor achtereenvolgens de subcommando's Rn, En, Hn, Ln, Gn en Fn. Start op punt x=50 en y=90. Neem voor n=20.

4-18:
Met welk subcommando kunnen we de kleur aangeven, waarin we een figuur willen laten tekenen? Wat is de default-waarde van de voorgrondkleur?

4-19:
Verander het programma uit 4-17 zodanig, dat het zeskant in de kleur zwart wordt getekend.

4-20:
Met welk subcommando kan een gespecificeerde figuur in een bepaalde stand worden geplaatst? Welke hoeken zijn mogelijk en wat zijn de bijbehorende waarden van n? Wat is de default waarde van n?

4-21:
Verander het programma uit vraag 4-19 zodanig, dat de gespecificeerde figuur 90 graden wordt gedraaid.

4-22:
Welk subcommando moet worden toegepast, wanneer we een figuur in een bepaalde schaal willen tekenen? Welke waarden kan n hierbij hebben?

4-23:
Wat stelt de schaalfactor precies voor? Hoe lang wordt voor het subcommando R20 de lijn, uitgedrukt in beeldpuntjes, wanneer de schaal n gelijk is aan 8. Wat is de default waarde voor de schaal en hoe groot is de daarbijbehorende schaalfactor?

4-24:
Verander het programma uit vraag 4-21 door middel van het subcommando Sn zodanig, dat de getrokken lijnen twee maal zo lang worden. Plaats hierbij de hoek (An), de schaal (Sn) en de kleur (Cn) in een apart DRAW-statement. Dit statement

moet voor het DRAW-statement, waarmee de figuur (zeskant) wordt getekend, komen te staan. Verander het startpunt in x=100 en y=90.

4-25:
Ken de hoek (An), de schaal (Sn) en de kleur (Cn) uit vraag

4-24 eerst toe aan een alfanumerieke variabele (A$) en pas vervolgens in het DRAW-statement, waarin de figuur wordt gespecificeerd, subcommando X toe voor het uitvoeren van de subcommando's An, Sn en Cn. Neem voor n respectievelijk de waarden 2 (hoek), 6 (schaal) en 3 (kleur).

4-26:
Hoe kan in een DRAW-statement de constante van een subcommando worden vervangen door een numerieke variabele. Wat is hiervoor het formaat?

4-27:
Verander het programma uit vraag 4-17 zodanig, dat de lengte van de lijnen (n) kan worden ingevoerd met een INPUT-statement. Gebruik vervolgens in de grafische mode 1 de numerieke variabele N voor de subcommando's. Na het indrukken van de RETURN-toets moet het programma weer terugspringen naar het INPUT-statement, opdat er een nieuwe waarde voor N kan worden ingevoerd. Neem als startpunt x=100 en y=150 en voor de hoek n=0.

5 Besturing van buitenaf

5-1:
In hoeveel posities kan de stuurknuppel van een joy-stick worden geplaatst en hoe kunnen de posities van de stuurknuppel worden opgevraagd?

5-2:
Met welke functie kan worden opgevraagd of een actieknop is ingedrukt?

5-3:
Wat geeft parameter n in de functie STICK(n) aan?

5-4:
Schrijf voor de grafische mode 1 een programma, waarmee de cursor-toetsen (pijltoetsen) als joy-stick worden geselecteerd. Verder dient bij het indrukken van:

- cursor "naar boven" DRAW "U1",
- cursor "naar rechts" DRAW "R1",
- cursor "naar beneden" DRAW "D1" en
- cursor "naar links" DRAW "L1" te worden uitgevoerd.

Wanneer er geen toets is ingedrukt, moet er worden teruggesprongen naar de regel waar de positie van de cursor-toetsen wordt afgevraagd (zie programma's in paragraaf 5.1). Laat het programma starten op regelnummer 150. Door het indrukken van de cursor-toetsen kunnen figuren op het scherm worden getekend.

5-5:
Met het programma uit vraag 5-4 kunnen we niet zelf bepalen waar we met tekenen zullen beginnen. Pas het programma nu zodanig aan, dat we met een INPUT-statement de beginpositie (coordinaten x en y) van de cursor kunnen invoeren en dat met een DRAW-statement de cursor op de gewenste positie wordt geplaatst.

5-6:
Pas het programma uit vraag 5-5 zodanig aan, dat we ook de kleur kunnen bepalen, waarin we de figuur willen tekenen. Het nummer van de kleur moet met een INPUT-statement worden ingevoerd. Het subcommando voor de kleurbepaling dient in het DRAW-statement voor de cursor-positionering te worden opgenomen.

5-7:
Wat geeft parameter n in de functie STRIG(n) aan?

5-8:
Met het programma uit vraag 5-6 zijn we niet in staat om delen van lijnen te onderbreken (niet te tekenen). Pas het programma nu zodanig aan, dat het mogelijk wordt een lijn te onderbreken, wanneer de spatiebalk wordt ingedrukt. Gebruik hiervoor de functie STRIG.
Wanneer de spatiebalk tegelijkertijd met een van de cursor-toetsen is ingedrukt, moet het betreffende subcommando BU1, BR1, BD1 of BL1 worden uitgevoerd. Wanneer de spatiebalk niet is ingedrukt, moet bij het indrukken van een van de cursor-toetsen het betreffende subcommando U1, R1, D1 of L1 worden uitgevoerd.

5-9:
Pas het programma uit vraag 5-8 zodanig aan, dat er, in plaats van met cursor-toetsen, kan worden getekend met een joy-stick, die is aangesloten op connector 1.

5-10:
Hoe kan door de spatiebalk of actieknop een programma worden onderbroken en vervolgens een bijbehorende subroutine worden aangeroepen?

5-11:
Met welk statement wordt het onderbreken van een programma, door het indrukken van de spatiebalk of actieknop, geactiveerd?

5-12:
Wat wordt er bij het aanroepen van een subroutine, als gevolg van het indrukken van een actieknop, door de computer automatisch uitgevoerd?

5-13:
Schrijf een programma, waarmee in de grafische mode 1 een lichtgroene lijn wordt getrokken vanuit positie x=0 en y=90 naar rechts. De lijn moet met het subcommando R1 worden getrokken en op positie x=150 stoppen. Hierna moet de lijn worden gewist en moet worden teruggesprongen naar het begin van het programma. Laat het programma op regelnummer 150 beginnen.

5-14:
Breid het programma uit vraag 5-13 zodanig uit, dat het opnieuw tekenen van de lijn op een willekeurig tijdstip begint. Gebruik hiervoor de programmaregel:

```
T=INT(RND(-TIME)*4000)
```

Pas T toe in een FOR...NEXT-lus voor het genereren van de vertraging. De vertraging in tijd moet in het programma na het statement CLS worden geplaatst.

5-15:
Breid het programma uit vraag 5-14 zodanig uit, dat door het indrukken van de spatiebalk het trekken van de lijn wordt gestopt. Het indrukken van de spatiebalk moet ook tot gevolg hebben, dat er naar een subroutine (regelnummer 300), die ervoor zorgt dat in tekstmode 1 op het beeldscherm de lengte van de lijn (X) wordt afgedrukt, wordt gesprongen.
Het activeren van de spatiebalk, door het statement STRIG(n) ON mag pas geschieden, nadat de lijn eenmaal volledig is

getekend en de tijdsvertraging is verstreken. Na het afdrukken van de tekst mag er pas worden teruggekeerd naar het hoofdprogramma, na het indrukken van de RETURN-toets.
Er moet worden teruggekeerd naar het begin van het programma en niet naar de plaats waar het programma eerder werd onderbroken. Dit betekent, dat achter het statement RETURN een regelnummer moet worden geplaatst.

5-16:
Pas het programma uit vraag 5-15 zodanig aan, dat de lijn kan worden gestopt met de actieknop op een joy-stick. Sluit hiervoor een joy-stick aan op connector 1.

5-17:
Pas het programma uit vraag 5-16 zodanig aan, dat de uitvoering van de subroutine niet wordt voortgezet door het indrukken van de RETURN-toets, maar door het in de bovenste positie (noord) plaatsen van de stuurknuppel van de joy-stick.

6 Lijnen, cirkels, vlakken en kleuren

6-1:
Met welke statements worden de grafische modes 1 en 2 ingesteld? Hoe is het beeldscherm ingedeeld voor de grafische mode 1 en hoe voor de grafische mode 2? Wat is in de grafische mode 2 de kleinste eenheid, waaruit figuren kunnen worden opgebouwd?

6-2:
Wanneer, in de grafische mode, de uitvoering van een programma is beeindigd, naar welke mode wordt dan teruggekeerd?

6-3:
Welke kleuren kunnen op het beeldscherm door het statement COLOR worden gedefinieerd? Geldt de randkleur voor alle modes? Zo niet, voor welke modes dan wel?

6-4:
Schrijf een programma, waarmee in de grafische mode 1 midden op het scherm een rechthoek wordt getekend van 80 beeldpuntjes lang bij 50 beeldpuntjes hoog. Gebruik hiervoor het

DRAW-statement. De achtergrondkleur moet hierbij wit zijn, de voorgrondkleur donkerrood en de randkleur lichtgroen.

6-5:
Wat is, in een programma, de juiste volgorde van het COLOR- en SCREEN-statement?

6-6:
Schrijf een programma, waarmee met het statement LINE in de grafische mode 1 een rechthoek wordt getekend van 80 beeldpuntjes lang en 50 beeldpuntjes hoog. Gebruik hiervoor vier LINE-statements zonder STEP. De kleur van de twee horizontale lijnen moet donkerrood zijn en de kleur van de twee verticale lijnen donkergeel. De achtergrondkleur en randkleur moeten wit zijn.

6-7:
Schrijf het programma uit vraag 6-6 opnieuw, maar nu met gebruikmaking van de faciliteit STEP.

6-8:
Verander het programma uit vraag 6-7 zodanig, dat het bruikbaar is voor de grafische mode 2.

6-9:
Hoe kan met 1 LINE-statement een rechthoek worden getekend?

6-10:
Verander het programma uit vraag 6-6 zodanig, dat er voor het tekenen van de rechthoek maar 1 LINE-statement nodig is. De kleur van de horizontale- en verticale lijnen moet donkerrood zijn.

6-11:
Welke kleur wordt er gebruikt, wanneer er in het LINE-statement geen kleurnummer is gedefinieerd?

6-12:
Verander het programma uit vraag 6-10 zodanig, dat het vlak van de rechthoek wordt ingekleurd met de kleur donkerblauw.

6-13:
Schrijf, voor de grafische mode 1, een programma, waarmee een cirkel wordt getekend met middelpunt x=128 en y=91, een straal van 60 beeldpuntjes en de kleur donkergroen. De voorgrondkleur en randkleur moeten wit zijn.

6-14:
Verander het programma uit vraag 6-13 zodanig, dat alleen het bovenste gedeelte van de cirkel wordt getekend.

6-15:
Verander het programma uit vraag 6-14 zodanig, dat het beginpunt en het eindpunt van de cirkelboog door rechte lijnen met het middelpunt worden verbonden.

6-16:
Schrijf voor de grafische mode 1 een programma, waarmee een ellips wordt getekend met middelpunt x=128 en y=91, een straal van 30 beeldpuntjes en een afplattingsfactor van 0,5. De voorgrondkleur moet wit zijn, de achtergrondkleur en randkleur donkerblauw. In het CIRCLE-statement moet geen kleurnummer worden gedefinieerd.

6-17:
Met welk statement kunnen vlakken worden ingekleurd? Geef ook het formaat van het statement.

6-18:
Schrijf voor de grafische mode 2 een programma, waarmee in het midden van het beeldscherm een rechthoek wordt getekend van 80 beeldpuntjes lang en 50 beeldpuntjes hoog. Gebruik voor het tekenen van de rechthoek een LINE-statement. De zijden van de rechthoek moeten in de kleur donkerblauw worden getekend. Vul met het statement PAINT het vlak van de rechthoek in met de kleur donkerrood. De voorgrond-, achtergrond- en randkleur moeten wit zijn

6-19:
Met welke statements kunnen punt voor punt figuren worden getekend en wat is het verschil tussen beide statements?

6-20:
Schrijf voor de grafische mode 1 een programma, waarmee van de functie SIN(X) een grafiek wordt getekend voor het interval 0.5pi tot en met 1.5pi. Neem als stapwaarde 0.03. De grafiek moet in de kleur zwart worden getekend. Laat het programma ook de coordinaatassen tekenen. De oorsprong van het assenstelsel bevindt zich op het beeldscherm in de positie x=40 en y=90. Vermenigvuldig, voor het verkrijgen van een beter beeld, X met een factor 30 en Y met een factor 60. Neem als voorgrondkleur zwart en als achtergrondkleur en randkleur wit.

6-21:
Pas het programma uit vraag 6-20 zodanig aan, dat de grafiek wordt getekend voor het interval 0.25pi tot en met 1.75pi.

6-22:
Met welke functie kan de kleur van een beeldpuntje worden afgevraagd? Hoe ziet het formaat van de functie er uit?

7 De macrotaal voor geluid

7-1:
Met welk statement wordt de macrotaal voor geluid aangeroepen?

7-2:
Waaruit bestaan de parameters, die bij het PLAY-statement worden gegeven?

7-3:
Op welke twee manieren kan een toon van een bepaalde hoogte worden aangegeven in het PLAY-statement?

7-4:
Hoeveel geluidskanalen heeft de geluidsprocessor die in MSX-computers wordt toegepast?

7-5:
Schrijf het statement, waarmee een toon op geluidskanaal A wordt geproduceerd. Gebruik hiervoor de toon C.

7-6:
Schrijf het statement, waarmee een toon op geluidskanaal B wordt geproduceerd.

7-7:
Schrijf het statement, waarmee zowel op geluidskanaal A als op geluidskanaal B een toon hoorbaar wordt gemaakt. Maak op elk kanaal een andere toon.

7-8:
Met welk subcommando kan de lengte van een noot worden bepaald?

7-9:
Met welk subcommando kan een octaaf worden geselecteerd?

7-10:
Schrijf een programma, waarmee de eerste twee regels van het liedje "klein klein kleutertje" worden gespeeld. Zet iedere regel in een apart PLAY-statement. Begin de eerst noot met de centrale C.

7-11:
Verander het programma uit vraag 7-10 zodanig, dat het tempo, waarin het liedje wordt gespeeld, met behulp van een PLAY-statement wordt bepaald.

7-12:
Met welk subcommando kunnen strings van subcommando's worden uitgevoerd?

7-13:
Wat is de functie van de puntkomma (;) in de subcommando-string?

7-14:
Schrijf de eerste twee regels van het liedje "klein klein kleutertje" nogmaals, doch zet de tonen nu in string-variabelen, die in de subcommando-string worden uitgevoerd.

7-15:
Welke twee instellingen zijn van belang voor het kiezen van een bepaalde soort van geluidsmodulatie?

7-16:
Voeg aan het programma uit vraag 7-14 een PLAY-statement toe, waarmee de tonen zodanig worden gemoduleerd, dat iedere toon op maximaal geluid begint. De tonen moeten nog net niet helemaal uitgestorven zijn op het moment dat de volgende toon begint.

7-17:
Schrijf het programma "klein klein kleutertje" nogmaals, doch voeg nu een statement toe waarmee het tempo kan worden ingegeven (met behulp van een INPUT-statement). De muziek dient vervolgens in het ingegeven tempo te worden gespeeld.

7-18:
Met welk subcommando kan een pauze tussen de tonen worden ingelast?

7-19:
Wat is de functie van een min-teken (-) in een subcommando-string?

7-20:
Wat is de functie van een punt (.) in een subcommando-string?

7-21:
Schrijf nu de eerste twee regels van het liedje "klein klein kleutertje nogmaals, doch gebruik nu het subcommando N.

7-22:
Met welke functie kunt u opvragen of de geluidsprocessor nog bezig is geluid te produceren?

7-23:
Wat is het resultaat van de functie PLAY(0), indien de geluidsprocessor nog geluid produceert via kanaal C?

7-24:
Wat is het resultaat van de functie PLAY(1), indien de geluidsprocessor geen geluid meer produceert via kanaal A?

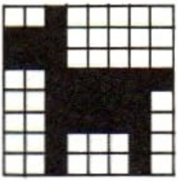
8 Sprites

8-1:
Hoe kunnen sprite-figuren worden vastgelegd?

8-2:
Welke sprite-groottes zijn mogelijk en hoe wordt de sprite-grootte gedefinieerd? Wat is foor de sprite-grootte de default-waarde?

8-3:
Met welke statement kunnen sprites op het beeldscherm worden geplaatst? Wat is het formaat van het statement?

8-4:
Schrijf voor de grafische mode 1 een programma, waarmee aan de alfanumerieke variabele S$ de informatie van de hiernaast gegeven figuur wordt toegekend. De figuur bestaat uit 8*8 beeldpuntjes. Gebruik een DATA-statement, waarin de informatie decimaal is vastgelegd. Ken, als laatste, variabele S$ toe aan systeemvariabele SPRITE$(0).



8-5:
Breid het programma uit vraag 8-4 zodanig uit, dat de sprite op het beeldscherm wordt geplaatst. De positie is x=128 en y=90, de kleur is zwart en het prioriteitsnummer is 0. De achtergrondkleur en randkleur zijn wit en de voorgrondkleur is zwart. De sprite-grootte is 0.

8-6:
Verander het programma uit vraag 8-5 zodanig, dat de sprite-grootte 1 wordt en er onder de hond een horizontale zwarte lijn over het beeldscherm wordt getekend. De hond komt dan op de zwarte lijn te staan.

8-7:
Breid het programma uit vraag 8-6 zodanig uit, dat de hond zich in horizontale zin over het beeldscherm verplaatst, van positie x=210 naar positie x=6. Gebruik hierbij een stapgrootte van -0.2.

8-8:
Breid het programma uit vraag 8-7 zodanig uit, dat op de achtergrond met het statement LINE een lichtgroen vlak van 30 beeldpuntjes lang en 16 beeldpuntjes hoog wordt getekend. De positie van de linker onderhoek van het vlak is x=128 en y=106.

8-9:
Schrijf een programma, waarmee in de grafische mode 1 de volgende acties plaatsvinden. Een zwarte hond (zie vraag 8-4) beweegt zich van rechts (positie x=240) naar links (positie x=-16) over het beeldscherm en vindt op positie x=60 een donkerblauw paaltje op zijn weg. Op het moment dat de hond het paaltje bereikt, stopt de hond en schiet er uit het paaltje een paarse kogel loodrecht omhoog. Op het moment dat de kogel de positie y=-16 bereikt, gaat de hond weer verder naar positie x=-16. Hierna beginnen de acties weer opnieuw. Alle bovengenoemde coordinaten slaan op de linker-

bovenhoeken van de sprites. Verdere gegevens zijn:



paaltje



kogel

a) De hond, het paaltje en de kogel moeten worden vastgelegd in sprites van 8*8. De informatie van de figuren moet direct worden vastgelegd in respectievelijk de systeemvariabelen SPRITE$(0), SPRITE$(1) en SPRITE$(2).
b) Voer het programma uit voor sprite-grootte 1 en grafische mode 1.
c) Neem voor de voorgrondkleur zwart en voor de achtergrondkleur en de randkleur wit.
d) Teken met het statement LINE een horizontale lijn over het beeldscherm op hoogte y=185 in de kleur donkergroen. Op deze lijn moet het paaltje worden geplaatst en de hond worden bewogen.
e) Neem voor de hond een stapgrootte van 0.2 en voor de kogel 0.5.
f) Geef het paaltje het hoogste prioriteitsnummer (0), de hond prioriteitsnummer 1 en de kogel nummer 2.
g) Start het programma op regelnummer 100 en de subroutine op regelnummer 300. In de subroutine wordt de kogel afgeschoten van positie y=170 naar positie y=-16. Vergeet in de subroutine niet SPRITE OFF te geven en in het hoofdprogramma weer SPRITE ON.

9 Wat is machinetaal?

9-1:
Welke microprocessor wordt in de MSX-computer gebruikt?

9-2:
Hoe zien machinetaalinstructies er uit?

9-3:
Wat is een assembler of assembleerprogramma?

9-4:
Wat is het principiele verschil tussen een BASIC-statement en een machinetaalinstructie?

9-5:
In welk microprocessorregister vinden alle berekeningen plaats?

9-6:
Welk microprocessorregister bevat het geheugenadres van de volgende machinetaalinstructie, die moet worden uitgevoerd?

9-7:
Met welk BASIC-statement kunnen we een stuk geheugen vrijmaken, zodat we daarin een machinetaalroutine kunnen laden?

9-8:
Welk commando dienen we te geven, om er achter te komen, hoe groot we het geheugengebied voor machinetaalroutines, mogen maken?

9-9:
Met welk statement kunnen we bepaalde waarden toekennen aan bepaalde geheugenadressen?

9-10:
Schrijf, nadat u uw computer, waarop geen disk is aangesloten, uit en weer aan hebt gezet, een programma waarmee u op de adressen 50000 en 50001 de decimale waarden 123 en 234 schrijft.

9-11:
Met welke functie kunt u bepaalde geheugenadressen uitlezen?

9-12:
Welk commando (statement in directe mode) dient u te geven om de inhoud van adres 50001 uit te lezen?

9-13:
Wat is het voordeel van het gebruik van hexadecimale getallen voor machinetaalprogramma's?

9-14:
Wat mag u nooit vergeten te doen, voordat u een nieuw ingetikt machinetaalprogramma gaat uitvoeren?

10 Aanroepen van machinetaalroutines

10-1:
In iedere MSX-computer zit een 32 kbytes ROM. Wat staat daar in? Is dat precies gelijk voor elke MSX-computer?

10-2:
Wat zijn BIOS entry points?

10-3:
Waarom kunnen de machinetaalinstructies IN en OUT beter niet worden gebruikt?

10-4:
Met welk BASIC-statement kan het startadres van een machinetaalroutine worden vastgelegd?

10-5:
Met welke functie kan een machinetaalroutine, waarvan het startadres eenmaal is vastgelegd, worden aangeroepen?

10-6:
Op welke twee manieren kan een machinetaalroutine uit de ROM via de BIOS entry point tabel worden aangeroepen?

10-7:
Op adres hexadecimaal "A2" staat het BIOS entry point, waarmee een teken naar het beeldscherm kan worden geschreven. Schrijf het BASIC-programma, waarmee dit entry point wordt aangeroepen.

10-8:
Om een door ons gewenst teken naar het scherm te schrijven, dient de accumulator van de Z80-microprocessor, voor het aanroepen van BIOS entry point A2, met de code van het gewenste teken te worden geladen. Dit kan alleen door middel van een machinetaalroutine. Deze machinetaalroutine is hierna gegeven:

adres:	inhoud:	
50000	&H3E	Laadt accumulator met
50001	&H41	hoofdletter A.
50002	&HCD	Roep het BIOS
50003	&HA2	entry point &HA2
50004	&H00	aan.
50005	&HC9	Keer terug naar BASIC.

Schrijf een BASIC-programma, waarmee bovenstaand machinetaalprogramma op de gegeven adressen in het geheugen wordt gezet en waarmee het machinetaalprogramma wordt aangeroepen.

10-9:
Wijzig het programma uit vraag 10-8 zodanig, dat het af te drukken teken via een INPUT-statement kan worden opgegeven. Maak hierbij geen gebruik van de standaard manier van het doorgeven van parameters van BASIC naar machinetaal.

10-10:
Op welke, standaard voorgeschreven, manier dient een waarde, die kleiner is dan 65536, van BASIC te worden doorgegeven aan een machinetaalprogramma?

10-11:
Welke soorten parameters kunnen tussen BASIC en machinetaal worden uitgewisseld?

10-12:
Wat is het resultaat van de functie VARPTR(variabelenaam)?

10-13:
Waartoe dienen de zogenaamde "systeemlocaties"?

10-14:
Wat is de functie van de "hook"-adressen?

10-15:
Waar bevinden zich de "hook"-adressen en wanneer worden deze adressen aangeroepen?

11 De programmeerbare geluidsprocessor

11-1:
Met behulp van welk statement kunnen we de programmeerbare geluidsgenerator direct besturen?

11-2:
Hoeveel besturingsregisters heeft de programmeerbare geluidsgenerator en wat wordt door ieder register bestuurd?

11-3:
Met welke waarde moet register 7 worden geladen, om het geluid en de ruis voor de kanalen A, B en C aan te zetten?

11-4:
Welke acties moeten achtereenvolgens worden genomen, om een toon te produceren met behulp van SOUND-statements?

11-5:
De inhoud van een registerpaar, dat de frequentie van de te produceren toon moet bepalen, dient gelijk te zijn aan de waarde 111860 gedeeld door de gewenste frequentie. Schrijf een programma, waarmee de inhoud van register 2 en 3 (voor kanaal B dus) de frequentie 1000 Herz weergeeft. Bereken de inhoud van beide registers zelf.

11-6:
Schrijf nu een programma, waarmee de inhoud van de toonhoogteregisters voor kanaal B wordt berekend en waarmee de betreffende registers worden geladen. De frequentie dient met behulp van een INPUT-statement te worden ingegeven.

11-7:
Vul het programma uit vraag 11-6 aan met een routine waarmee het volume van kanaal B wordt ingesteld. Ook het volume dient met behulp van een INPUT-statement te worden ingegeven.

11-8:
Completeer het programma uit vraag 11-7 met een statement waarmee kanaal B wordt aangeschakeld. Zorg er vervolgens voor, dat er een nieuwe frequentie en een nieuw geluidsniveau kan worden ingesteld, waarna deze nieuwe toon te gehore wordt gebracht.

11-9:
Wat is de hoogste ruisfrequentie en hoe kan de gewenste ruisfrequentie worden bepaald?

11-10:
Schrijf een programma, waarmee via kanaal C een ruis met een frequentie van 55930 Herz en een maximaal volume ten gehore wordt gebracht. Start het programma met een statement, waarmee alle geluids- en ruiskanalên worden uitgeschakeld.

11-11:
Schrijf een programma, waarmee de geluidsfrequentie en de ruisfrequentie worden opgevraagd, waarna beide frequenties via kanaal A met elkaar worden gemengd en ten gehore gebracht. Stel het volume voor kanaal A op maximaal in.

11-12:
Welke acties dienen achtereenvolgens te worden genomen, om geluiden te moduleren?

11-13:
De periodetijd van de modulatie wordt bepaald, door de waarde 6991 te vermenigvuldigen met de gewenste periodetijd in seconden en het verkregen resultaat in de registers 11 en 12 te laden. Schrijf een programma, waarmee de periodetijd op 2 seconden wordt ingesteld. Bereken de inhoud van de registers 11 en 12 in het programma.

11-14:
Schrijf een programma, waarmee de modulatievorm en de periodetijd van de modulatie via INPUT-statements wordt opgevraagd, waarna de gewenste modulatie wordt ingesteld. Om het resultaat te kunnen horen, dient voor kanaal A een toon te worden gegenereerd, en dient de modulatie voor kanaal A te worden geactiveerd.

11-15:
Schrijf een programma, waarmee een sirene wordt nagebootst. Gebruik hiervoor twee FOR...NEXT-lussen die binnen een derde FOR...NEXT-lus worden uitgevoerd. De derde lus dient om de eerste twee een aantal malen te herhalen. De eerste lus dient het geluid in toonhoogte te laten stijgen, terwijl de tweede lus de toonhoogte laat dalen.

12 De Video Display Processor

12-1:
Waarvoor dient de Video Display Processor?

12-2:
Hoe groot is het videogeheugen van de VDP?

12-3:
Hoeveel besturingsregisters heeft de VDP?

12-4:
Door middel van welke systeemvariabele zijn de tabellen in de VDP toegankelijk?

12-5:
Hoe kan er informatie uit het videogeheugen worden gelezen en hoe kan informatie in het vidogeheugen worden geschreven?

12-6:
Hoe kan de inhoud van de besturingsregisters van de VDP worden uitgelezen en hoe kan er nieuwe informatie in de besturingsregisters worden geplaatst?

12-7:
Kan er, door de gebruiker, in alle registers van de VDP informatie worden geschreven?

12-8:
Welke tabellen worden in de tekstmode 1 gebruikt en wat zijn de beginadressen?

12-9:
Wat is het eerste adres van het blok van 8 bytes in de matrix-tabel voor de letter S?

12-10:
Schrijf een programma, waarmee van de letter S de enen en nullen van 8 bytes uit de matrix-tabel op het scherm worden afgedrukt. Noteer na het uitvoeren van het programma de inhoud van de 8 bytes.

12-11:
Schrijf een programma, waarmee in de matrix-tabel de 8 bytes voor de letter S als hiernaast aangegeven worden veranderd. Voer, door middel van de functie INKEY$, een aantal keren de letter S in, om te zien wat het resultaat is.



12-12:
Verander het programma uit vraag 12-11 zodanig, dat in de 8 bytes ten behoeve van de letter S weer de oorspronkelijke informatie komt te staan. Gebruik hiervoor de genoteerde informatie uit vraag 12-10.

12-13:
Welke tabellen worden in de tekstmode 2 gebruikt en wat zijn de beginadressen?

12-14:
Schrijf een programma, waarmee de kleuren van de tekens met de ASCII-codes 72 tot en met 79 wordt veranderd in donker-rood (voorgrondkleur) en wit (achtergrondkleur). Type, na het uitvoeren van het programma de betreffende tekens in.

12-15:
Verander het programma uit vraag 12-14 zodanig, dat de oorspronkelijke kleuren weer worden verkregen.

12-16:
Schrijf voor de tekstmode 1 een programma, waarmee in het betreffende besturingsregister in de VDP de voorgrondkleur wordt veranderd in paars en de achtergrondkleur in licht-groen.

Antwoorden en uitwerkingen

1 Standaard MSX-functies en eigen functies

1-1:
Met de functie ABS(X).

1-2:

```
10 CLS
20 INPUT "Waarde X";X
30 INPUT "Waarde Y";Y
40 PRINT
50 PRINT ABS(X-Y)
60 PRINT
70 GOTO 20
```

1-3:
Met de functie FIX(X) worden zowel voor positieve- als negatieve getallen alle cijfers achter de decimale punt afgekapt.
Met de functie INT(X) worden voor positieve getallen alle cijfers achter de decimale punt afgekapt. Voor negatieve getallen wordt naar beneden afgerond, dat wil zeggen, dat INT(-32.17)=-33 en FIX(-32.17)=-32.

1-4:

```
10 CLS
20 INPUT "Waarde X";X
30 PRINT
40 G=FIX(X)
50 B=X-FIX(X)
60 PRINT G;SPC(6);B
70 PRINT
80 GOTO 20
```

1-5:
De ASCII-code voor Carriage Return is 13. In een programma kan Carriage Return worden geactiveerd met PRINT CHR$(13).

1-6:

```
10 CLS
20 INPUT "Variabele a";A
30 INPUT "Variabele b";B
40 INPUT "Variabele c";C
50 PRINT
```

```
60 Y=SQR((A+B)/C)
70 PRINT Y
80 I$=INKEY$:IF I$<>CHR$(13) GOTO 80
90 GOTO 10
```

1-7:
Met de functie RND(X) kunnen willekeurige getallen worden gegenereerd.

1-8:

```
10 CLS
20 A=RND(-TIME)
30 FOR I=1 TO 12
40 PRINT INT(RND(1)*22);
50 NEXT I
60 PRINT
70 END
```

1-9:

```
10 CLS
20 A=RND(-TIME)
30 FOR I=1 TO 12
40 PRINT INT(RND(1)*21+1);
50 NEXT I
60 PRINT
70 END
```

1-10:

```
10 CLS
15 INPUT "Grootste getal";G
20 A=RND(-TIME)
30 FOR I=1 TO 12
40 PRINT INT(RND(1)*G+1);
50 NEXT I
60 I$=INKEY$:IF I$<>CHR$(13) GOTO 60
70 GOTO 10
```

1-11:

```
10 CLS
15 INPUT "Grootste getal";G
16 INPUT "Aantal getallen";T
20 A=RND(-TIME)
30 FOR I=1 TO T
40 PRINT INT(RND(1)*G+1);
50 NEXT I
60 I$=INKEY$:IF I$<>CHR$(13) GOTO 60
70 GOTO 10
```

1-12:
Met het statement DEF FN.

1-13:
DEF FNOPDR(B,H)=.5*B*H

1-14:

```
10 DEF FNOPDR(B,H)=.5*B*H
20 CLS
30 INPUT "Basis  B";B
40 INPUT "Hoogte H";H
50 PRINT
60 PRINT "Oppervlakte =";FNOPDR(B,H)
70 I$=INKEY$:IF I$<>CHR$(13) GOTO 70
80 GOTO 20
```

1-15:

```
10 DEF FNINHC(R,H)=(3.14*R^2)*H
20 CLS
30 INPUT "Straal R cilinder";R
40 INPUT "Hoogte H cilinder";H
50 PRINT
60 PRINT "Inhoud cilinder =";FNINHC(R,H)
70 I$=INKEY$:IF I$<>CHR$(13) GOTO 70
80 GOTO 20
```

1-16:
De variabelen worden ook wel dummy-variabelen genoemd.

1-17:
De variabelen in een DEF-statement behoeven niet dezelfde namen te hebben als in de uit te voeren functie.

1-18:

```
10 DFF FNFUN(X,Y,Z)=SQR((X+Y)^2/Z)
20 CLS
30 INPUT "Variabelen A,B,C";A,B,C
40 PRINT
50 PRINT "R =";FNFUN(A,B,C)
60 I$=INKEY$:IF I$<>CHR$(13) GOTO 60
70 GOTO 20
```

2 Afdrukken op scherm en papier

2-1:
Met het commando LLIST kan het programma dat in het geheugen staat worden afgedrukt op papier.

2-2:
Met het commando LLIST. (een punt achter LLIST) wordt de laatst behandelde programmaregel afgedrukt op papier.

2-3:
```
KEY 2,"LLIST"+CHR$(13)
```

2-4:
```
100 CLS
110 INPUT "Eerste getal ";E
120 INPUT "Laatste getal";L
130 LPRINT CHR$(13)
140 FOR I=E TO L
150 LPRINT USING "####  #######  #.####";I,I^2,1/I
160 NEXT I
170 PRINT
180 PRINT "Tabel is afgedrukt."
190 END
```

2-5:
De systeemvariabele CSRLIN geeft als resultaat het nummer van de regel (y-coordinaat), waarop de cursor zich, op het moment van uitvoering van de functie, bevindt. De inhoud van CSRLIN kan 0 tot en met 23 zijn, waarbij 0 de bovenste regel van het scherm voorstelt.

2-6:
De systeemvariabele CSRLIN kan alleen in de tekstmodes 1 en 2 worden toegepast.

2-7:
```
100 CLS
110 INPUT "Beginwaarde B";B
120 PRINT USING "####  ########";B,B^2
130 B=B+1
140 IF CSRLIN<22 GOTO 120
150 I$=INKEY$:IF I$<>CHR$(13) GOTO 150
160 CLS
170 GOTO 120
```

2-8:
De systeemvariabele POS(0) geeft als resultaat de momentele positie van de cursor op de regel (x-coordinaat). Het resultaat kan 0 tot en met 39 zijn.

2-9:
```
10 CLS
20 FOR I=65 TO 90
30 PRINT CHR$(I);SPC(2);
40 IF POS(0)>20 THEN PRINT
50 NEXT I
60 PRINT
70 END
```

2-10:
De systeemvariabele LPOS(0) geeft een indicatie van de momentele print-positie in het print-buffer.

2-11:

```
10 LPRINT CHR$(13)
20 FOR I=65 TO 90
30 LPRINT CHR$(I);SPC(2);
40 IF LPOS(0)>20 THEN LPRINT
50 NEXT I
60 PRINT "Letters zijn afgedrukt."
70 END
```

3 Goniometrische functies

3-1:
MSX-BASIC kent voor de goniometrie de volgende standaard functies: SIN(X), COS(X), TAN(X) en ATN(X).

3-2:
Het argument X kan als volgt worden uitgedrukt:

a. Als numerieke constante.
b. Als numerieke variabele.
c. Als uitdrukking.

3-3:
Het aantal radialen X=G*3.14159/180, waarin G het aantal graden voorstelt.

3-4:

```
100 CLS
110 INPUT "Graden";G
120 PRINT
130 X=G*3.14159/180
140 PRINT "Het aantal radialen is";X
150 I$=INKEY$:IF I$<>CHR$(13) GOTO 150
160 GOTO 100
```

3-5:

```
100 CLS
110 PRINT "BEREKENEN COSINUS VAN EEN HOEK"
120 PRINT
130 GOSUB 300
140 PRINT "Cosinus van";G;"graden is ";
150 PRINT USING "##.#####";COS(X)
160 I$=INKEY$:IF I$<>CHR$(13) GOTO 160
```

```
170 GOTO 100
300 INPUT "Graden";G
310 X=G*3.14159/180
320 RETURN
```

3-6:

```
300 INPUT "Hoek in gr,min,sec";G,M,S
310 X=(G+M/60+S/360)*3.14159/180
320 RETURN
```

3-7:

```
100 CLS
110 PRINT "SINUS- EN COSINUSTABEL"
120 PRINT
130 INPUT "Beginwaarde";B
140 INPUT "Eindwaarde ";E
150 PRINT
160 FOR G=B TO E
170 X=G*3.14159/180
180 PRINT USING "###";G;
190 PRINT SPC(3);
200 PRINT USING "##.#####";SIN(X);
210 PRINT SPC(3);
220 PRINT USING "##.#####";COS(X)
230 IF CSRLIN<20 GOTO 260
240 I$=INKEY$:IF I$<>CHR$(13) GOTO 240
250 CLS
260 NEXT G
270 END
```

3-8:
ATN(a/b)=X en
TAN(X)=a/b.
Een hoek, uitgedrukt in radialen, wordt als volgt in graden omgezet: Y=(G*180)/3.14159
Hierin stelt G het aantal radialen voor en Y het aantal graden.

3-9:

```
100 CLS
110 INPUT "Zijde a";A
120 INPUT "Zijde b";B
130 PRINT
140 R=ATN(A/B)
150 G=R*180/3.14159
160 PRINT "Hoek A is";
170 PRINT USING "###.##";G;
180 PRINT " graden."
190 I$=INKEY$:IF I$<>CHR$(13) GOTO 190
200 GOTO 100
```

3-10:
Met de functie EXP(X) wordt de X-de macht van het grondt van de natuurlijke logaritme e berekend.

3-11:

```
100 CLS
110 PRINT "A = 2.718282 ^ X"
120 PRINT
130 PRINT " X";SPC(10);"A"
140 FOR X=.1 TO 1.9 STEP .1
150 PRINT USING "#.#";X;
160 PRINT SPC(6);
170 PRINT USING "#.#####";EXP(X)
180 NEXT X
190 END
```

3-12:
Met de functie LOG(X) kunnen we de natuurlijke logaritn (grondtal e) van argument X berekenen. Het argument X ka als volgt worden uitgedrukt:

a. Als numerieke constante.
b. Als numerieke variabele.
c. Als uitdrukking.

3-13:
LOG10(X)=0.434294*LOG(X)

3-14:

```
200 CLS
210 PRINT "BEREKENING NATUURLIJKE LOGARITME"
220 PRINT
230 INPUT "Getal X";X
240 PRINT
250 PRINT "LOG(X) = ";LOG(X)
260 RETURN
```

3-15:

```
200 CLS
210 PRINT "BEREKENING NATUURLIJKE LOGARITME"
220 PRINT
230 INPUT "Getal X";X
240 PRINT
250 PRINT "LOG(X) = ";LOG(X)
260 RETURN
300 CLS
310 PRINT "BEREKENING BRIGGSE LOGARITME"
320 PRINT
330 PRINT "Getal X";X
340 PRINT
350 PRINT "LOG10(X) = ";.434294*LOG(X)
360 RETURN
```

3-16:

```
100 CLS
105 PRINT "BEREKENEN VAN LOGARITMEN"
110 PRINT:PRINT
115 PRINT "1. Natuurlijke logaritme":PRINT
120 PRINT "2. Briggse logaritme":PRINT
125 PRINT "3. Programma beeindiging"
130 PRINT:PRINT:PRINT
135 INPUT "Keuze";K
140 IF K<1 OR K>3 THEN PRINT "Verkeerde
keuze!":GOTO 135
145 IF K=3 THEN END
150 ON K GOSUB 200,300
155 I$=INKEY$:IF I$<>CHR$(13) GOTO 155
160 GOTO 100
200 CLS
210 PRINT "BEREKENING NATUURLIJKE LOGARITME"
220 PRINT
230 INPUT "Getal X";X
240 PRINT
250 PRINT "LOG(X) = ";LOG(X)
260 RETURN
300 CLS
310 PRINT "BEREKENING BRIGGSE LOGARITME"
320 PRINT
330 INPUT "Getal X";X
340 PRINT
350 PRINT "LOG10(X) = ";.434294*LOG(X)
360 RETURN
```

4 De grafische macrotaal

4-1:
In de grafische macrotaal wordt gebruik gemaakt van het statement DRAW.

4-2:
Het statement DRAW mag alleen worden gebruikt in de grafische modes 1 (SCREEN 2) en 2 (SCREEN 3).

4-3:
In de grafische mode 1 is het beeldscherm verdeeld in 256 keer 192 beeldpuntjes. Elk beeldpuntje wordt bepaald door een x-coordinaat (0-255) en een y-coordinaat (0-191). De

indeling van het scherm in de grafische mode 2 is precies hetzelfde als in de grafische mode 1. Het verschil met de grafische mode 1 is echter, dat de figuren op het scherm in plaats van uit beeldpuntjes, worden opgebouwd uit blokjes, welke bestaan uit 4 keer 4 beeldpuntjes.

4-4:
De kleinste eenheid in de grafische mode 1 is 1 beeldpuntje en in de grafische mode 2 een blokje van 4 keer 4 beeldpuntjes. Het positioneren kan in beide modes op 1 beeldpuntje nauwkeurig.

4-5:
Het subcommando Mx,y kan worden gebruikt voor het trekken van een lijn.

4-6:
Subcommando B. Met de volgende twee subcommando's wordt de cursor linksonder op het beeldscherm geplaatst: BM0,191.

4-7:

```
10 SCREEN 2
20 DRAW "BM0,191M255,0"
30 GOTO 30
```

4-8:

```
10 SCREEN 2
20 DRAW "BM0,191M255,0BM0,0M255,191"
30 GOTO 30
```

4-9:
Met het subcommando N wordt de cursor weer teruggeplaatst naar het beginpunt.

4-10:
Daar een grafisch beeldscherm ook als beeldscherm wordt beschouwd, dient het eerst te worden geopend, alvorens er gegevens naartoe kunnen worden geschreven. Het openen geschiedt met het volgende statement: OPEN "GRP:" AS #X.

4-11:

```
10 SCREEN 2
15 OPEN "GRP:" AS #1
20 DRAW "BM0,191M255,0BM0,0M255,191"
30 DRAW "BM127,48":PRINT #1,"A"
40 DRAW "BM192,96":PRINT #1,"B"
50 DRAW "BM127,144":PRINT #1,"C"
60 DRAW "BM64,96":PRINT #1,"D"
70 GOTO 70
```

4-12:

```
10 SCREEN 2
20 DRAW "BM80,60M170,60M170,120M80,120M80,60"
30 GOTO 30
```

4-13:
In geval van relatieve waarden, worden x en y voorafgegaan door een plusteken (+) of een minteken (-).

cursorpositie	subcommando
x=40, y=70	M120,90
	M+80,-20

4-14:

```
10 SCREEN 2
20 DRAW "BM80,60M+90,+0M+0,+60M-90,+0M+0,-60"
30 GOTO 30
```

4-15:
Met de volgende subcommando's kunnen lijnen in 8 verschillende richtingen worden getrokken:
Un, Dn, Ln, Rn, En, Fn, Gn en Hn.

4-16:

```
10 SCREEN 2
20 DRAW "BM128,91NU60ND60NL60NR60NE60NF60NG60NH60"
30 GOTO 30
```

4-17:

```
10 SCREEN 2
20 DRAW "BM50,90R20E20H20L20G20F20"
30 GOTO 30
```

4-18:
Met het subcommando Cn kunnen we de kleur aangeven, waarin we een figuur willen tekenen. De default-waarde voor de kleur is wit.

4-19:

```
10 SCREEN 2
20 DRAW "C1BM50,90R20E20H20L20G20F20"
30 GOTO 30
```

4-20:
Met het subcommando An kan een gespecificeerde figuur in een bepaalde stand worden geplaatst. A0 = 0 graden.
A1 = 90 graden.
A2 = 180 graden.
A3 = 270 graden.
De default-waarde is 0 graden.

4-21:

```
10 SCREEN 2
20 DRAW "C1BM50,90A1R20E20H20L20G20F20"
30 GOTO 30
```

4-22:
Subcommando Sn dient om een figuur in een bepaalde schaal te tekenen. n kan een waarde hebben van 1 tot en met 255.

4-23:
De schaalfactor is gelijk aan n gedeeld door 4. De lijn wordt 20x8:4=40 beeldpuntjes lang. De default-waarde voor de schaal is 4, hetgeen betekent, dat de schaalfactor 1 is.

4-24:

```
10 SCREEN 2
20 DRAW "A1S8C1"
30 DRAW "BM100,90R20E20H20L20G20F20"
40 GOTO 40
```

4-25:

```
10 SCREEN 2
20 A$="A2S6C3"
30 DRAW "XA$;BM100,90R20E20H20L20G20F20"
40 GOTO 40
```

4-26:
Alle constanten (x, y en n) in de subcommando's kunnen worden vervangen door numerieke variabelen. Het formaat ziet er als volgt uit: subcommando = numerieke variabele ;

4-27:

```
10 CLS
20 INPUT "Lengte n";N
30 SCREEN 2
40 DRAW "A0BM100,150R=N;E=N;H=N;L=N;G=N;F=N;"
50 I$=INKEY$:IF I$<>CHR$(13) GOTO 50
60 GOTO 20
```

5 Besturing van buitenaf

5-1:
De stuurknuppel van een joy-stick kan in 8 posities worden geplaatst. De posities kunnen worden opgevraagd met de functie STICK(n).

5-2:
Met de functie STRIG(n) kan worden afgevraagd of een actie-knop is ingedrukt.

5-3:
Parameter n in de functie STICK geeft aan of de cursortoetsen als joy-stick (n=0) zijn geselecteerd, of de joy-stick op connector 1 (n=1) of op joy-stick 2 (n=2).

5-4:

```
150 SCREEN 2
160 P=STICK(0)
170 IF P=0 GOTO 160
180 IF P=1 THEN DRAW "U1":GOTO 160
190 IF P=3 THEN DRAW "R1":GOTO 160
200 IF P=5 THEN DRAW "D1":GOTO 160
210 IF P=7 THEN DRAW "L1":GOTO 160
220 GOTO 160
```

5-5:

```
100 CLS
110 INPUT "Coordinaten x en y";X,Y
150 SCREEN 2
155 DRAW "BM=X;,=Y;"
160 P=STICK(0)
170 IF P=0 GOTO 160
180 IF P=1 THEN DRAW "U1":GOTO 160
190 IF P=3 THEN DRAW "R1":GOTO 160
200 IF P=5 THEN DRAW "D1":GOTO 160
210 IF P=7 THEN DRAW "L1":GOTO 160
220 GOTO 160
```

5-6:

```
100 CLS
110 INPUT "Coordinaten x en y";X,Y
120 INPUT "Kleur         (0-15)";K
150 SCREEN 2
155 DRAW "C=K;BM=X;,=Y;"
160 P=STICK(0)
170 IF P=0 GOTO 160
180 IF P=1 THEN DRAW "U1":GOTO 160
190 IF P=3 THEN DRAW "R1":GOTO 160
200 IF P=5 THEN DRAW "D1":GOTO 160
210 IF P=7 THEN DRAW "L1":GOTO 160
220 GOTO 160
```

5-7:
Parameter n in de functie STRIG(n) geeft het volgende aan:
n=0 : de spatiebalk is als actieknop geselecteerd.
n=1 of 3: de eerste, respectievelijk tweede, actieknop van de joy-stick, die is aangesloten op connector 1, is geselecteerd.

n=2 of 4: de eerste, respectievelijk tweede, actieknop van de joy-stick, die is aangesloten op connector 2, is geselecteerd.

5-8:

```
100 CLS
110 INPUT "Coordinaten x en y";X,Y
120 INPUT "Kleur        (0-15)";K
150 SCREEN 2
155 DRAW "C=K;BM=X;,=Y;"
160 P=STICK(0)
165 S=STRIG(0)
170 IF P=0 GOTO 160
175 IF S=-1 GOTO 211
180 IF P=1 THEN DRAW "U1":GOTO 160
190 IF P=3 THEN DRAW "R1":GOTO 160
200 IF P=5 THEN DRAW "D1":GOTO 160
210 IF P=7 THEN DRAW "L1":GOTO 160
211 IF P=1 THEN DRAW "BU1":GOTO 160
212 IF P=3 THEN DRAW "BR1":GOTO 160
213 IF P=5 THEN DRAW "BD1":GOTO 160
214 IF P=7 THEN DRAW "BL1":GOTO 160
220 GOTO 160
```

5-9:

```
100 CLS
110 INPUT "Coordinaten x en y";X,Y
120 INPUT "Kleur        (0-15)";K
150 SCREEN 2
155 DRAW "C=K;BM=X;,=Y;"
160 P=STICK(1)
165 S=STRIG(1)
170 IF P=0 GOTO 160
175 IF S=-1 GOTO 211
180 IF P=1 THEN DRAW "U1":GOTO 160
190 IF P=3 THEN DRAW "R1":GOTO 160
200 IF P=5 THEN DRAW "D1":GOTO 160
210 IF P=7 THEN DRAW "L1":GOTO 160
211 IF P=1 THEN DRAW "BU1":GOTO 160
212 IF P=3 THEN DRAW "BR1":GOTO 160
213 IF P=5 THEN DRAW "BD1":GOTO 160
214 IF P=7 THEN DRAW "BL1":GOTO 160
220 GOTO 160
```

5-10:
Met behulp van het statement ON STRIG GOSUB.

5-11:
Met het statement STRIG(n) ON.

5-12:
Het statement STRIG(n) STOP.

5-13:

```
150 SCREEN 2
160 DRAW "C3BM0,90"
170 X=0
180 DRAW "R1"
190 X=X+1
200 IF X<150 GOTO 180
210 CLS
220 GOTO 150
```

5-14:

```
150 SCREEN 2
160 DRAW "C3BM0,90"
170 X=0
180 DRAW "R1"
190 X=X+1
200 IF X<150 GOTO 180
210 CLS
215 T=INT(RND(-TIME)*4000)
216 FOR I=1 TO T:NEXT I
220 GOTO 150
```

5-15:

```
100 ON STRIG GOSUB 300
150 SCREEN 2
160 DRAW "C3BM0,90"
170 X=0
180 DRAW "R1"
190 X=X+1
200 IF X<150 GOTO 180
210 CLS
215 T=INT(RND(-TIME)*4000)
216 FOR I=1 TO T:NEXT I
217 STRIG(0) ON
220 GOTO 150
300 SCREEN 0
310 PRINT "U heeft gereageerd na";X;"pixels."
320 I$=INKEY$:IF I$<>CHR$(13) GOTO 320
330 STRIG(0) OFF
340 RETURN 150
```

5-16:

```
100 ON STRIG GOSUB ,300
150 SCREEN 2
160 DRAW "C3BM0,90"
170 X=0
180 DRAW "R1"
190 X=X+1
200 IF X<150 GOTO 180
210 CLS
```

```
215 T=INT(RND(-TIME)*4000)
216 FOR I=1 TO T:NEXT I
217 STRIG(1) ON
220 GOTO 150
300 SCREEN 0
310 PRINT "U heeft gereageerd na";X;"pixels."
320 I$=INKEY$:IF I$<>CHR$(13) GOTO 320
330 STRIG(1) OFF
340 RETURN 150
```

5-17:

```
100 ON STRIG GOSUB ,300
150 SCREEN 2
160 DRAW "C3BM0,90"
170 X=0
180 DRAW "R1"
190 X=X+1
200 IF X<150 GOTO 180
210 CLS
215 T=INT(RND(-TIME)*4000)
216 FOR I=1 TO T:NEXT I
217 STRIG(1) ON
220 GOTO 150
300 SCREEN 0
310 PRINT "U heeft gereageerd na";X;"pixels."
320 P=STICK(1):IF P<>1 GOTO 320
330 STRIG(1) OFF
340 RETURN 150
```

6 Lijnen, cirkels, vlakken en kleuren

6-1:
De grafische modes 1 en 2 worden respectievelijk ingesteld met SCREEN 2 en SCREEN 3. Zie voor de beeldschermindelingen de oplossingen van de vragen 4-3 en 4-4.

6-2:
Naar de tekstmode.

6-3:
De volgende kleuren kunnen door het statement COLOR worden gedefinieerd:

a. Voorgrondkleur.
b. Achtergrondkleur.

c. Randkleur.

De randkleur geldt alleen voor de tekstmode 2 en de grafische modes 1 en 2.

6-4:

```
10 COLOR 6,15,3
20 SCREEN 2
30 DRAW "BM87,70D50R80U50L80"
40 GOTO 40
```

6-5:

Het COLOR-statement moet in een programma voor het SCREEN-statement staan.

6-6:

```
10 COLOR 15,15,15
20 SCREEN 2
30 LINE (88,71)-(168,71),6
40 LINE (168,71)-(168,121),10
50 LINE (168,121)-(88,121),6
60 LINE (88,121)-(88,71),10
70 GOTO 70
```

6-7:

```
10 COLOR 15,15,15
20 SCREEN 2
30 LINE (88,71)-STEP(+80,+0),6
40 LINE -STEP(+0,+50),10
50 LINE -STEP(-80,+0),6
60 LINE -STEP(+0,-50),10
70 GOTO 70
```

6-8:

```
10 COLOR 15,15,15
20 SCREEN 3
30 LINE (88,71)-STEP(+80,+0),6
40 LINE -STEP(+0,+50),10
50 LINE -STEP(-80,+0),6
60 LINE -STEP(+0,-50),10
70 GOTO 70
```

6-9:

Met het statement LINE (x1,y1)-(x2,y2),K,B.
Wanneer de letter B achter het statement wordt geplaatst, wordt er een rechthoek getekend met de gegeven lijn als diagonaal.

6-10:

```
10 COLOR 15,15,15
20 SCREEN 2
30 LINE (88,71)-(168,121),6,B
40 GOTO 40
```

6-11:
Wanneer geen kleurnummer (voorgrondkleur) wordt gespecificeerd in het LINE-statement, zal de laatst gedefinieerde voorgrondkleur worden gebruikt.

6-12:
```
10 COLOR 15,15,15
20 SCREEN 2
30 LINE (88,71)-(168,121),4,BF
40 GOTO 40
```

6-13:
```
10 COLOR 15,15,15
20 SCREEN 2
30 CIRCLE (128,91),60,12
40 GOTO 40
```

6-14:
```
10 COLOR 15,15,15
20 SCREEN 2
30 CIRCLE (128,91),60,12,6.28,3.14
40 GOTO 40
```

6-15:
```
10 COLOR 15,15,15
20 SCREEN 2
30 CIRCLE (128,91),60,12,-6.28,-3.14
40 GOTO 40
```

6-16:
```
10 COLOR 15,4,4
20 SCREEN 2
30 CIRCLE (128,91),30,,,,.5
40 GOTO 40
```

6-17:
Met het statement PAINT(x,y),Z kunnen in de grafische mode 1 vlakken worden ingekleurd. Met het statement PAINT(x,y),Z,R kunnen in de grafische mode 2 vlakken worden ingekleurd.

6-18:
```
10 COLOR 15,15,15
20 SCREEN 3
30 LINE (80,70)-(160,120),4,B
40 PAINT (120,95),6,4
50 GOTO 50
```

6-19:
Met de statements PSET(x,y),Z en PRESET(x,y),Z kunnen figuren punt voor punt worden getekend. Wanneer in het statement PRESET het kleurnummer (Z) niet is gegeven, wordt de achtergrondkleur, die op dat moment actief is, als kleur genomen. Wanneer in het statement PSET het kleurnummer (Z) niet is gegeven, wordt de voorgrondkleur, die op dat moment actief is, als kleur genomen.

6-20:

```
10 COLOR 1,15,15
20 SCREEN 2
30 LINE (0,90)-(255,90),1
40 LINE (40,0)-(40,192),1
50 FOR X=1.57 TO 4.61 STEP .03
60 Y=SIN(X)
70 PSET (40+X*30,90-Y*60),1
80 NEXT X
90 GOTO 90
```

6-21:

```
10 COLOR 1,15,15
20 SCREEN 2
30 LINE (0,90)-(255,90),1
40 LINE (40,0)-(40,192),1
50 FOR X=.785 TO 5.495 STEP .03
60 Y=SIN(X)
70 PSET (40+X*30,90-Y*60),1
80 NEXT X
90 GOTO 90
```

6-22:
Met de functie POINT(x,y) kan de kleur van een beeldpuntje worden afgevraagd.

7 De macrotaal voor geluid

7-1:
Met het PLAY-statement, dat wordt gevolgd door een of meerdere parameters.

7-2:
De parameters bestaan uit subcommando's.

7-3:
Door de noot (C, D, E, F, G, A of B) aan te geven, eventueel aangevuld met de gewenste octaaf, of door het toonnummer (0 tot en met 95) aan te geven.

7-4:
Drie geluidskanalen, kanaal A, B en C.

7-5:

```
10 PLAY "C"
```

7-6:

```
10 PLAY "","C"
```

7-7:

```
10 PLAY "C","E"
```

7-8:
Met het subcommando L. Achter L wordt de lengte aangegeven. Default is de lengte 4. Een hogere waarde achter L maakt een kortere noot, een lagere waarde een hogere noot.

7-9:
Met het subcommando O. Achter O wordt het gewenste octaaf aangegeven. De mogelijke octaven zijn van 1 tot en met 8.

7-10:

```
10 PLAY "CEL8GGGFEEDDL4C"
20 PLAY "O3GO4L8CDEFGGGFEEDDL4C"
```

7-11:

```
10 PLAY "T255"
20 PLAY "CEL8GGGFEEDDL4C"
30 PLAY "O3GO4L8CDEFGGGFEEDDL4C"
```

7-12:
Met het subcommando X kan de inhoud van een stringvariabele worden uitgevoerd, alsof die string de subcommandostring zelf is.

7-13:
De puntkomma (;) heeft als functie, het einde van de naam van de variabele, die door een X-subcommando wordt uitgevoerd, aan te geven.

7-14:

```
10 PLAY "T255"
20 A1$="CEL8GGGFEEDDL4C"
30 A2$="O3GO4L8CDEFGGGFEEDDL4C"
40 PLAY "XA1$;XA2$;"
```

7-15:
De vorm (subcommando S van shape) en de periodetijd (subcommando M van modulatietijd).

7-16:

```
10 PLAY "T150"
20 A1$="CEL8GGGFEEDDL4C"
30 A2$="O3GO4L8CDEFGGGFEEDDL4C"
40 PLAY "M5000S8"
50 PLAY "XA1$;XA2$;"
```

7-17:

```
10 INPUT "Het gewenste tempo";T
20 PLAY "T=T;"
20 A1$="CEL8GGGFEEDDL4C"
30 A2$="O3GO4L8CDEFGGGFEEDDL4C"
```

```
40 PLAY "M5000S8"
50 PLAY "XA1$;XA2$;"
```

7-18:
Met het subcommando R wordt een rust gecreeerd. De lengte van de rust wordt op dezelfde manier aangegeven als de lengte van een toon.

7-19:
Door het minteken (-) achter een noot te zetten, kan de verlaagde toon van die hoot worden gegenereerd. Dit komt overeen met de zwarte toets van een pianoklavier, die links van de aangegeven witte toets zit. (bijvoorbeeld de zwarte toets links van de G-toets is de toon Ges, hetgeen wordt aangegeven met het subcommando G-.)

7-20:
Met de punt (.) wordt de toon die vlak voor die punt staat met de helft verlengd.

7-21:

```
10 PLAY "L4N36N40L8N43N43N43N41N40N40N38
N38L4N36"
20 PLAY "L4N31L8N36N38N40N41N43N43N43N41
N40N40N38N38L4N36"
```

7-22:
Met de functie PLAY(X), waarbij de parameter X het te onderzoeken kanaal aangeeft.

7-23:
PLAY(0) onderzoekt de status van alle drie de kanalen. Indien kanaal C nog geluid produceert, zal het resultaat van de functie -1 zijn.

7-24:
PLAY(1) onderzoekt de status van kanaal A. Indien kanaal A geen geluid meer produceert, zal het resultaat van de functie 0 zijn.

8 Sprites

8-1:
Figuren (sprites) kunnen worden vastgelegd in de systeemva-

8-2:
De sprite-grootte wordt gedefinieerd in het SCREEN-statement met een van de waarden 0, 1, 2 of 3.

0 = kleine sprites, 8*8 beeldpuntjes (default-waarde).
1 = kleine sprites, vergroot tot 16*16 beeldpuntjes.
2 = grote sprites, 16*16 beeldpuntjes.
3 = grote sprites, vergroot tot 32*32 beeldpuntjes.

8-3:
Met het volgende statement kunnen sprites op het beeldscherm worden geplaatst: PUT SPRITE P,(x,y),K,X

Waarin:
- P = prioriteitsnummer (0-31).
- x,y = coordinaten van de positie waar de sprite moet worden geplaatst.
- K = kleurnummer (0-15).
- X = sprite-nummer.

8-4:

```
10 SCREEN 2,0
20 S$=""
30 FOR I=1 TO 8
40 READ A
50 S$=S$+CHR$(A)
60 NEXT I
70 SPRITE$(0)=S$
80 DATA 32,224,224,63,62,62,34,34
```

8-5:

```
5 COLOR 1,15,15
10 SCREEN 2,0
20 S$=""
30 FOR I=1 TO 8
40 READ A
50 S$=S$+CHR$(A)
60 NEXT I
70 SPRITE$(0)=S$
75 PUT SPRITE 0,(128,90),1,0
76 GOTO 76
80 DATA 32,224,224,63,62,62,34,34
```

8-6:

```
5 COLOR 1,15,15
10 SCREEN 2,1
20 S$=""
30 FOR I=1 TO 8
40 READ A
50 S$=S$+CHR$(A)
60 NEXT I
70 SPRITE$(0)=S$
71 LINE (0,106)-(255,106)
```

```
80 DATA 32,224,224,63,62,62,34,34
```

8-7:

```
5 COLOR 1,15,15
10 SCREEN 2,1
20 S$=""
30 FOR I=1 TO 8
40 READ A
50 S$=S$+CHR$(A)
60 NEXT I
70 SPRITE$(0)=S$
71 LINE (0,106)-(255,106)
74 FOR X=210 TO 6 STEP -.2
75 PUT SPRITE 0,(X,90),1,0
76 NEXT X
77 GOTO 77
80 DATA 32,224,224,63,62,62,34,34
```

8-8:

```
5 COLOR 1,15,15
10 SCREEN 2,1
20 S$=""
30 FOR I=1 TO 8
40 READ A
50 S$=S$+CHR$(A)
60 NEXT I
70 SPRITE$(0)=S$
71 LINE (0,106)-(255,106)
72 LINE (128,90)-(158,106),3,BF
74 FOR X=210 TO 6 STEP -.2
75 PUT SPRITE 0,(X,90),1,0
76 NEXT X
77 GOTO 77
80 DATA 32,224,224,63,62,62,34,34
```

8-9:

```
100 COLOR 1,15,15
110 SCREEN 2,1
115 ON SPRITE GOSUB 300
116 SPRITE ON
120 SPRITE$(0)=CHR$(32)+CHR$(224)+CHR$(2
24)+CHR$(63)+CHR$(62)+CHR$(62)+CHR$(34)+
CHR$(34)
130 SPRITE$(1)=CHR$(24)+CHR$(24)+CHR$(24
)+CHR$(24)+CHR$(24)+CHR$(24)+CHR$(24)+CH
R$(24)
140 SPRITE$(2)=CHR$(0)+CHR$(0)+CHR$(0)+C
HR$(24)+CHR$(24)+CHR$(0)+CHR$(0)+CHR$(0)
150 LINE (0,185)-(255,185),12
160 PUT SPRITE 0,(60,169),4,1
```

```
170 FOR X=240 TO -16 STEP -.2
180 PUT SPRITE 1,(X,169),1,0
190 NEXT X
195 SPRITE ON
200 GOTO 160
300 FOR Y=170 TO -16 STEP -.5
310 PUT SPRITE 2,(60,Y),13,2
320 NEXT Y
325 SPRITE OFF
330 RETURN
```

9 Wat is machinetaal?

9-1:
De Z80 microprocessor wordt in alle MSX-computers gebruikt.

9-2:
Machinetaalinstructies bestaan uit een aantal (veelvoud van 8) bitjes. We kunnen de binaire waarde schrijven als een binair-, een decimaal-, een octaal- of hexadecimaal getal.

9-3:
Een assembler is een programma, dat ons in staat stelt machinetaalinstructies met behulp van voor ons begrijpbare woorden in te voeren. De assembler zet deze door ons ingegeven woorden om naar echte machinetaalinstructies, die uit nulletjes een eentjes bestaan. Ieder door ons ingegeven woord is een machinetaal instructie.

9-4:
Een BASIC-statement wordt door de BASIC-interpreter of BASIC-compiler omgezet naar een aantal machinetaalinstructies. Een BASIC-statement kan worden vertaald naar machinetaalinstructies, een machinetaalinstructie kan door de microprocessor worden uitgevoerd.

9-5:
In de accumulator, ofwel register A, vinden alle berekeningen plaats.

9-6:
Het registerpaar PC (Program Counter) bevat het adres van de

volgende machinetaalinstructie.

9-7:
Met het CLEAR-statement kan het hoogste adres, dat door BASIC mag worden gebruikt, worden aangegeven. In een standaard MSX-computer strekt het gebied dat voor BASIC toegankelijk is, zich uit tot &HF379. Door met het CLEAR-statement een lager adres aan te geven, bijvoorbeeld &HF000, wordt het gebied tussen &HF000 en &HF380 vrijgemaakt voor gebruik door eigen machinetaalroutines.

9-8:
Met het commando PRINT FRE(0) geeft ons de grootte van het nog niet gebruikte geheugengebied op. Weten we hoe groot dit niet gebruikte geheugengebied is, dan kunnen we met CLEAR een aantal bytes reserveren voor onze eigen machinetaalroutines.

9-10:

```
10 CLEAR 200,49999!
20 POKE 50000!,123
30 POKE 50001!,234
```

9-11:
Met de functie PEEK(X), waarbij X het uit te lezen geheugenadres voorstelt.

9-12:
Het geheugenadres 50001 kan worden uitgelezen met het commando: PRINT PEEK(50001).

9-13:
Enerzijds blijft er, door het gebruik van hexadecimale getallen, een directe relatie met de binaire waarde bestaan, anderzijds is het veel gemakkelijker om twee hexadecimale cijfers te schrijven en te onthouden dan acht binaire cijfers.

9-14:
Schrijf elk programma, waarin machinetaal wordt gebruikt, eerst weg naar cassette, voordat u het programma voor het eerst uitvoert. Mocht er een typefout in de machinetaalroutine zitten, dan is de kans groot dat er een "hangup"-situatie zal ontstaan, met als gevolg het verloren gaan van het gehele programma.

10 Aanroepen van machinetaalroutines

10-1:
In de ROM zit een groot aantal machinetaalroutines. Bovendien zit er een tabel met beginadressen van een groot aantal machinetaalroutines in. De machinetaalroutines kunnen onderling verschillen. De tabel is voor elke MSX-computer dezelfde, dat wil zeggen, dat er dezelfde routines in dezelfde volgorde in staan aangegeven. De startadressen van die routines zin machine-afhankelijk, doch dat is voor ons van minder belang.

10-2:
BIOS entry point zijn adressen (uit een tabel in de ROM, die voor elke MSX-computer dezelfde is), die we kunnen aanroepen om gebruik te maken van machinetaalroutines die al in de ROM aanwezig zijn.

10-3:
Bij de machinetaalinstructies IN en OUT dienen we een poortnummer op te geven. Deze poortnummers vormen geen deel van de MSX-standaard. Zouden we toch gebruik van poortnummers maken, dan zouden we de kans lopen om programma's te schrijven, die niet op alle MSX-computers werken.

10-4:
Met het statement DEFUSRX=NNNN. Hierin is X het volgnummer van de machinetaalroutine en NNNN het startadres van de machinetaalroutine.

10-5:
Met de functie USRX(Y). Hierin is X het volgnummer van de machinetaalroutine. Dit volgnummer werd met het statement DEFUSR aan de machinetaalroutine toegekend. Y is een parameter, die aan de machinetaalroutine wordt doorgegeven. Deze parameter mag een numerieke of stringvariabele zijn.

10-6:
Direct vanuit BASIC, of vanuit een machinetaalroutine. Indien de routine die via een BIOS entry point wordt aangeroepen geen parameters nodig heeft of oplevert, dan is de directe manier de beste.

10-7:

```
10 DEFUSR0=&HA2
20 A=USR0(0)
```

10-8:

```
10 CLEAR 200,49999!
20 POKE 50000!,&H3E
30 POKE 50001!,&H41
40 POKE 50002!,&HCD
50 POKE 50003!,&HA2
60 POKE 50004!,&H0
70 POKE 50005!,&HC9
80 DEFUSR0=50000!
90 A=USR0(0)
100 GOTO 90
```

10-9:

```
10 CLEAR 200,49999!
20 INPUT "AF TE DRUKKEN LETTER";I$
30 I=ASC(I$)
40 POKE 50000!,&H3E
50 POKE 50001!,I
60 POKE 50002!,&HCD
70 POKE 50003!,&HA2
80 POKE 50004!,&H0
90 POKE 50005!,&HC9
100 DEFUSR0=50000!
110 A=USR0(0)
120 GOTO 110
```

10-10:
De waarde wordt in een integer-variabele gezet. Deze integer-variabele wordt in de USR-functie opgenomen. De machinetaalroutine moet nu de integer-waarde opzoeken, door eerst te controleren of op adres &HF663 te kijken of daar de waarde 2 staat. Is dit inderdaad het geval, dan staat de integer-waarde op de adressen &HF7F8 en &HF7F9. Adres &HF7F9 bevat het meest significante deel van de integer-waarde.

10-11:
Er kunnen vier soorten parameters worden doorgegeven tussen BASIC en machinetaal, namelijk:

- integer-waarden
- strings
- waarden met enkelvoudige nauwkeurigheid
- waarden met dubbele nauwkeurigheid

10-12:
Het resultaat van de functie VARPTR(variabelenaam) is, dat het geheugenadres van de aangegeven variabele wordt verkregen.

10-13:
De systeemlocaties worden gevormd door een deel van het RAM-

geheugen. Deze locaties van het RAM worden gebruikt om er de variabele gegevens voor de ROM-routines in op te slaan. In de ROM kan immers niet worden geschreven.

10-14:
Het doel van een "hook"-adres is, de programmeur in staat te stellen nieuwe functies aan de bestaande ROM-routines toe te voegen.

10-15:
De "hook"-adressen staan in het geheugengebied dat wordt aangegeven met systeemlocaties. Een "hook"-adres wordt door de ROM-routines aangeroepen. In een groot aantal ROM-routines wordt een "hook"-adres aangeroepen. Na initialiseren van de computer (na aanschakelen) staan er op die "hook"-adressen echter alleen RET-instructies, waardoor het aanroepen geen effect zal hebben. De RET-instructies kunnen echter door de programmeur worden vervangen door een spronginstructie naar een eigen machinetaalroutine.

11 De programmeerbare geluidsprocessor

11-1:
Met het statement SOUND R,I. Hierin is R het besturingsregister en I de inhoud die we aan dat register willen toekennen.

11-2:
Er zijn in totaal 14 registers, die als volgt worden gebruikt:

```
0 en 1   - de toonhoogte van kanaal A
2 en 3   - de toonhoogte van kanaal B
4 en 5   - de toonhoogte van kanaal C
6        - de ruisfrequentie
7        - het aan- en uitschakelen van kanalen
8        - het volume van kanaal A
9        - het volume van kanaal B
10       - het volume van kanaal C
11 en 12 - de periodetijd voor de modulator
13       - de modulatievorm
```

11-3:
De bitjes voor de ruis moeten op 0 worden gezet, terwijl alle andere bits op 1 blijven staan (0 = aan, 1 = uit). Dit levert de waarde 128+64+4+2+1 op, hetgeen 199 is.

11-4:
Eerst moeten de toonhoogte registers worden geladen, daarna moet het volume worden bepaald, en ten slotte dient het betreffende kanaal te worden aangeschakeld.

11-5:
```
10 SOUND 2,112
20 SOUND 3,0
```

11-6:
```
10 INPUT "Gewenste frequentie";F
20 RP=INT(111860!/F)
30 R3=RP\256
40 R2=RP-256*R3
50 SOUND 2,R2
60 SOUND 3,R3
```

11-7:
```
10 INPUT "Gewenste frequentie";F
20 INPUT "Gewenst volume";V
30 RP=INT(111860!/F)
40 R3=RP\256
50 R2=RP-256*R3
60 SOUND 2,R2
70 SOUND 3,R3
80 SOUND 9,V
```

11-8:
```
10 INPUT "Gewenste frequentie";F
20 INPUT "Gewenst volume";V
30 RP=INT(111860!/F)
40 R3=RP\256
50 R2=RP-256*R3
60 SOUND 2,R2
70 SOUND 3,R3
80 SOUND 9,V
90 SOUND 7,253
100 GOTO 10
```

11-9:
De hoogste ruisfrequentie is 111860 Herz. Deze frequentie wordt gedeeld door de waarde, die in register 6 wordt geladen. Het resultaat van deze deling is de werkelijke ruisfrequentie.

11-10:
```
10 SOUND 7,255
20 SOUND 6,2
30 SOUND 10,15
```

```
40 SOUND 7,223
```

11-11:

```
10 INPUT "Gewenste ruisfrequentie";R
20 INPUT "Gewenste toonhoogte";T
30 RF=INT(111860!/R)
40 IF RF>31 THEN RF=31
50 RP=INT(111860!/T)
60 R1=RP\256
70 R0=RP-256*R1
80 SOUND 0,R0
90 SOUND 1,R1
100 SOUND 6,RF
110 SOUND 8,15
120 SOUND 7,246
```

11-12:
Eerst dient de modulatievorm te worden gekozen, vervolgens de periodetijd van de modulatie en tenslotte dient de modulatie te worden aangeschakeld (de waarde 16 in de volume-registers).

11-13:

```
10 PT=2*6991
20 MSR=PT\256
30 LSR=PT-256*MSR
40 SOUND 11,LSR
50 SOUND 12,MSR
60 SOUND 8,16
70 SOUND 13,8
80 SOUND 7,254
```

11-14:

```
10 INPUT "Modulatievorm (0-15)";MV
20 INPUT "Periodetijd (seconden)";T
30 PT=T*6991
40 MSR=PT\256
50 LSR=PT-256*MSR
60 SOUND 11,LSR
70 SOUND 12,MSR
80 SOUND 1,2
90 SOUND 8,16
100 SOUND 13,MV
110 SOUND 7,254
```

11-15:

```
10 SOUND 8,15
20 SOUND 7,254
30 FOR J=1 TO 10
40 FOR I=200 TO 1000 STEP 5
50 RP=INT(111860!/I)
60 R1=RP\256
70 R0=RP-256*R1
```

```
80 SOUND 0,R0
90 SOUND 1,R1
100 NEXT I
110 FOR I=1000 TO 200 STEP -10
120 RP=INT(111860!/I)
130 R1=RP\256
140 R0=RP-256*R1
150 SOUND 0,R0
160 SOUND 1,R1
170 NEXT I
180 NEXT J
```

12 De Video Display Processor

12-1:
De Video Display Processor zorgt voor de besturingssignalen en informatie, die nodig zijn voor het produceren van de beelden (teksten, figuren, etc.) op het scherm.

12-2:
Het videogeheugen van de VDP is 16 kilobytes groot.

12-3:
De VDP heeft 9 besturingsregisters.

12-4:
Door middel van de systeemvariabele BASE(X) zijn de tabellen in de VDP toegankelijk.

12-5:
Informatie lezen met de functie VPEEK. Informatie schrijven met het statement VPOKE.

12-6:
Met de systeemvariabele VDP(X) kan er informatie worden gelezen uit de besturingsregisters en informatie worden geschreven in de besturingsregisters.

12-7:
Nee. Register 8 mag alleen worden gelezen.

12-8:
In de tekstmode 1 wordt van twee tabellen gebruik gemaakt,

namelijk van de scherm-info-tabel, met beginadres BASE(0), en de matrix-tabel, met beginadres BASE(2).

12-9:
Het eerste adres is: BASE(2)+83*8=BASE(2)+664.

12-10:

```
10 CLS
20 A=BASE(2)+664
30 FOR I=A TO A+7
40 A$="00000000"+BIN$(VPEEK(I))
50 PRINT RIGHT$(A$,8)
60 NEXT I
70 PRINT
80 END
```

12-11:

```
100 VPOKE BASE(2)+664,&H38
110 VPOKE BASE(2)+665,&H28
120 VPOKE BASE(2)+666,&HEE
130 VPOKE BASE(2)+667,&H82
140 VPOKE BASE(2)+668,&HEE
150 VPOKE BASE(2)+669,&H28
160 VPOKE BASE(2)+670,&H38
170 VPOKE BASE(2)+671,&H0
180 I$=INKEY$:IF I$="" GOTO 180
190 PRINT I$;
200 GOTO 180
```

12-12:

```
100 VPOKE BASE(2)+664,&H70
110 VPOKE BASE(2)+665,&H88
120 VPOKE BASE(2)+666,&H80
130 VPOKE BASE(2)+667,&H70
140 VPOKE BASE(2)+668,&H8
150 VPOKE BASE(2)+669,&H88
160 VPOKE BASE(2)+670,&H70
170 VPOKE BASE(2)+671,&H0
180 I$=INKEY$:IF I$="" GOTO 180
190 PRINT I$;
200 GOTO 180
```

12-13:
In de tekstmode 2 wordt van drie tabellen gebruik gemaakt, namelijk van de scherm-info-tabel, met beginadres BASE(5), de matrix-tabel, met beginadres BASE(7), en de kleur-tabel, met beginadres BASE(6). Wanneer met sprites wordt gewerkt, wordt behalve van de drie hiervoor genoemde tabellen ook nog gebruik gemaakt van een sprite-info-tabel met beginadres BASE(8) en een SPRITE$-tabel met beginadres BASE(9).

12-14:

```
10 SCREEN 1
20 VPOKE BASE(6)+9,&H6F
30 END
```

12-15:

```
10 SCREEN 1
20 VPOKE BASE(6)+9,&HF4
30 END
```

12-16:

```
10 SCREEN 0
20 VDP(7)=&HD3
30 END
```

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Akkermans, Wessel

MSX leerboek / Wessel Akkermans, Piet den Heijer. – Oosterend: Stark-Texel
Opdrachten bij
Dl. 2
ISBN 90 6398 556 8
SISO 365.3 UDC 681.3
Trefw.: MSX (computer)

1e druk 1985
ISBN 90 6398 556 8

uitgeverij STARK-TEXEL

postbus 302 - 1794 ZG Oosterend tel. 02223 - 661